Loan/EMI Payment on time

In the partial fulfillment for the award of BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE & ENGINEERING

Submitted by

Abhishek Kumar Rai Regd. No. GCS1740592
Satish Kaushik Regd. No. GCS1740603
Vikrant Kumar Mall Regd. No. GCS1740578

Under the Guidance of

Mr. Manoj Kumar Sachan
Head of Department
(CSE Department)

Sant Longowal Institute of Engineering & Technology, Longowal – 148106
District – Sangrur, Punjab August, 2019

Index

INTRODUCTION

Like all private businesses Bank makes money from service charges and fees. These fees vary based on the products, ranging from account fees (monthly maintenance charges, minimum balance fees, overdraft fees, non-sufficient funds (NSF) charges), safe deposit box fees, and late fees. Many loan products also contain fees in addition to interest charges. Banks also earn money from interest they earn by lending out money to other clients. The funds they lend comes from customer deposits. However, the interest rate paid by the bank on the money they borrow is less than the rate charged on the money they lend but the ‘Non-performing Assets’ (NPA) are increasing very rapidly which in turn lock the money and neither the customer get lots of offer nor the insurance company makes profit.

Going by latest financial numbers, LIC mirrors the bank-like blunder in doling out loans to private sector entrepreneurs. The LIC's gross NPAs at 6.10 per cent for the first six months (April-September) of 2019-20 are comparable to banks such as YES Bank, Axis Bank and ICICI Bank.
The private sector lenders, once known for best asset quality, saw rising NPAs due to challenging operating environment. In the second quarter of 2019-20, YES Bank ended with gross NPAs of 7.39 percent , ICICI Bank with 6.37 per cent and Axis Bank with 5.03 per cent.
The state-owned insurance behemoth also lends to corporate sector by way of term loans and non-convertible debentures (NCDs).
The corporation with total assets of over Rs 36 lakh crore has its own share of woes trusting the private sector entrepreneurs. The LIC has reported total gross NPAs of around Rs 30,000 crore as on September 30, 2019.

Therefore, to prevent from these type of situations when the money is struck in form of NPA we have created a deep learning model to predict whether a customer will pay the loan EMI or not.

PROPOSED WORK

We have prepared a deep learning model through tensorflow and sklearn.

Code:

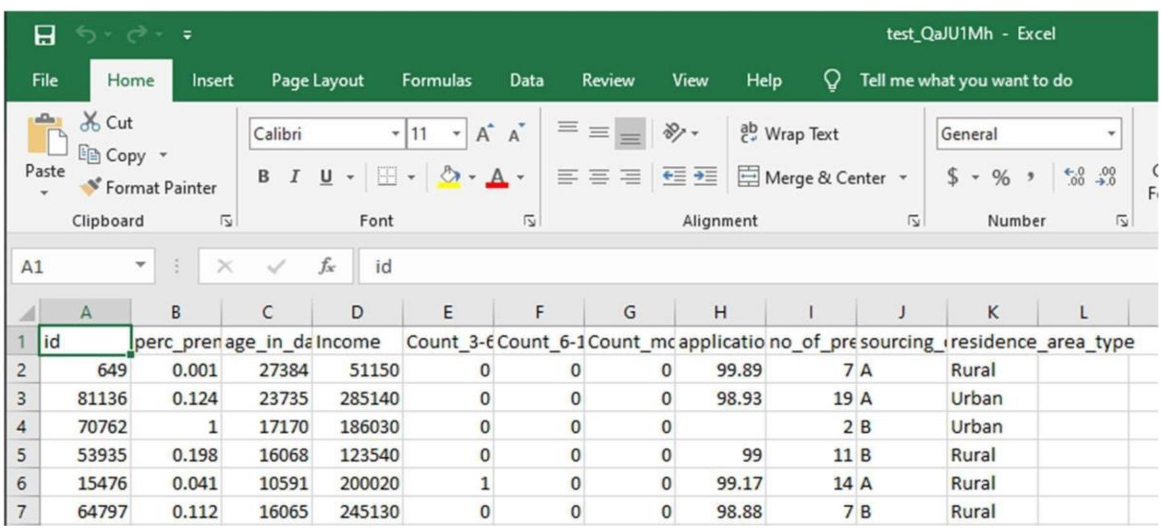
```
# -*- coding: utf-8 -*- """loan.ipynb

Automatically generated by Colaboratory.

Original file is located at
   https://colab.research.google.com/drive/1lY9pVfPMKIobTt78NtSflhL64LOUut_9 """

# Commented out IPython magic to ensure Python compatibility.
# %pip install jupyterthemes

# Commented out IPython magic to ensure Python compatibility.
import pandas as pd from sklearn.preprocessing import
StandardScaler, MinMaxScaler
from sklearn.model_selection import train_test_split
import numpy as np import pandas as pd import matplotlib.pyplot as plt
import seaborn as sns from jupyterthemes import jtplot jtplot.style(theme =
'monokai', context = 'notebook', ticks = True, grid = False)
# %matplotlib inline

data=pd.read_csv('/content/drive/MyDrive/Data/Loan-emi/train.csv')
data2=pd.read_csv('/content/drive/MyDrive/Data/Loan-emi/test.csv')

df_date = data.groupby(by='age_in_days').mean() df_date

#data.hist(bins=30, figsize=(20,20), color = 'r')
#sns.pairplot(data)

corr_matrix = data.corr() plt.figure(figsize
= (12,12)) sns.heatmap(corr_matrix,
annot=True)

data=data.fillna(method='ffill')
X=data[['perc_premium_paid_by_cash_credit','age_in_days','Count_3-
6_months_late','Count_612_months_late','Count_more_than_12_months_late','application_unde
rwriting_score','no_of_pr emiums_paid','sourcing_channel','residence_area_type']].values
```

```
y=data['target'].values
y=y.reshape(-1,1) print(X.shape)
data2=data2.fillna(method='ffill')

from sklearn import preprocessing
sou_cha = preprocessing.LabelEncoder()

X[:,7] = sou_cha.fit_transform(X[:,7])  res_are
= preprocessing.LabelEncoder()
X[:,8] = res_are.fit_transform(X[:,8])

from sklearn import preprocessing sou_cha
= preprocessing.LabelEncoder() X1[:,7] =
sou_cha.fit_transform(X1[:,7])  res_are =
preprocessing.LabelEncoder()
X1[:,8] = res_are.fit_transform(X1[:,8])

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()  X_scale =
scaler.fit_transform(X)
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X1_scale = scaler.fit_transform(X1)

from sklearn.model_selection import train_test_split
X_train, X_val_and_test, y_train, y_val_and_test = train_test_split(X_scale,y,test_size=0.2)
X_val, X_test, y_val, y_test = train_test_split(X_val_and_test, y_val_and_test, test_size=0.5)

model = Sequential([
   Dense(250, activation='relu', input_shape=(9,)),
   Dense(100, activation='relu'),
   Dense(10, activation='relu'),
Dense(1, activation='sigmoid')])
model.compile(optimizer='sgd',
loss='binary_crossentropy',
metrics=['accuracy']) hist =
model.fit(X_train, y_train,
batch_size=32, epochs=2,
          validation_data=(X_val, y_val))

print('\033[91m'"Accuracy on training set:{:.3f}".format(model.evaluate(X_train,y_train)[1]))
print('\033[93m'"Accuracy on Validation set:{:.3f}".format(model.evaluate(X_val,y_val)[1]))
print('\033[92m'"Accuracy on test set:{:.3f}".format(model.evaluate(X_test,y_test)[1]))
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import mean_squared_error, accuracy_score

lr_model = LogisticRegression()
lr_model.fit(X_train,y_train)

lr_accuracy = lr_model.score(X_test, y_test) lr_accuracy

plt.plot(hist.history['loss']) plt.title('Model loss
Progress During Training') plt.xlabel('Epoch')
plt.ylabel('Training Loss')
plt.legend(['Training Loss'])

from sklearn.tree import DecisionTreeClassifier dt_model
= DecisionTreeClassifier()
dt_model.fit(X_train, y_train)

dt_accuracy = dt_model.score(X_test, y_test) dt_accuracy

from sklearn.ensemble import RandomForestClassifier rf_model1 =
RandomForestClassifier(n_estimators=400, max_depth=10)
rf_model1.fit(X_train, y_train)

rf_accuracy = rf_model1.score(X_test, y_test) rf_accuracy

y_predict = rf_model1.predict(X_test)
plt.plot(y_test, y_predict, '^' , color = 'r')

#scaler = StandardScaler()
#scaler_y = scaler.fit(y)
#y_predict_orig = scaler_y.inverse_transform(y_predict)
#y_test_orig = scaler_y.inverse_transform(y_test)

k = X_test.shape[1]
n = len(X_test) n

from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error from
math import sqrt
rsme = float(format(np.sqrt(mean_squared_error(y_test,y_predict)),'.3f'))
mse  = mean_squared_error(y_test, y_predict) mae
= mean_absolute_error(y_test, y_predict)
r2   = r2_score(y_test, y_predict)
adj_r2 = 1-((1-r2)*(n-1))/(n-k-1)
```

```
print('RSME =',rsme)
print('MSE =',mse)
print('MAE =',mae) print('R2
=',r2) print('ADJ_R2
=',adj_r2)

from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score, auc

y_score      =   model.predict_proba(X_test)[:,0]
y_score1 = lr_model.predict_proba(X_test)[:,1]
y_score2 = dt_model.predict_proba(X_test)[:,1]
y_score3 = rf_model.predict_proba(X_test)[:,1]

fpr,  tpr,  th = roc_curve(y_test, y_score) fpr1,
tpr1, th1 = roc_curve(y_test, y_score1) fpr2,
tpr2, th2 = roc_curve(y_test, y_score2) fpr3,
tpr3, th3 = roc_curve(y_test, y_score3)

print('roc_auc_score for Artificial Neural Network: ', roc_auc_score(y_test, y_score))
print('roc_auc_score for Logistic Regression: ', roc_auc_score(y_test, y_score1))
print('roc_auc_score for DecisionTree: ', roc_auc_score(y_test, y_score2))
print('roc_auc_score for RandomForest: ', roc_auc_score(y_test, y_score3))

plt.subplots(1, figsize=(10,10)) plt.title('Receiver Operating
Characteristic - Artificial Neural Network') plt.plot(fpr, tpr) plt.plot([0,
1], ls="--")
plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate') plt.show()

plt.subplots(1, figsize=(10,10))
plt.title('Receiver Operating Characteristic - Logistic Regression')
plt.plot(fpr1, tpr1) plt.plot([0, 1], ls="--")
plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate') plt.show()

plt.subplots(1, figsize=(10,10)) plt.title('Receiver
Operating Characteristic - DecisionTree')
plt.plot(fpr2, tpr2) plt.plot([0,
1], ls="--")
plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")
```

```
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate') plt.show()

plt.subplots(1, figsize=(10,10)) plt.title('Receiver Operating
Characteristic - RandomForest')
plt.plot(fpr3, tpr3) plt.plot([0,
1], ls="--")
plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate') plt.show().
```



id	perc_pren	age_in_da	Income	Count_3-6	Count_6-1	Count_mo	applicatio	no_of_pre	sourcing_	residence_area_type
649	0.001	27384	51150	0	0	0	99.89	7	A	Rural
81136	0.124	23735	285140	0	0	0	98.93	19	A	Urban
70762	1	17170	186030	0	0	0		2	B	Urban
53935	0.198	16068	123540	0	0	0	99	11	B	Rural
15476	0.041	10591	200020	1	0	0	99.17	14	A	Rural
64797	0.112	16065	245130	0	0	0	98.88	7	B	Rural

Test Datasets

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	id	perc_pren	age_in_da	Income	Count_3-6	Count_6-1	Count_mc	applicatio	no_of_pre	sourcing_	residence	premium	target
2	110936	0.429	12058	355060	0	0	0	99.02	13	C	Urban	3300	1
3	41492	0.01	21546	315150	0	0	0	99.89	21	A	Urban	18000	1
4	31300	0.917	17531	84140	2	3	1	98.69	7	C	Rural	3300	0
5	19415	0.049	15341	250510	0	0	0	99.57	9	A	Urban	9600	1
6	99379	0.052	31400	198680	0	0	0	99.87	12	B	Urban	9600	1
7	59951	0.54	17527	282080	2	0	0	99.18	9	B	Rural	22200	1

Train Datasets

```
print('\033[91m'"Accuracy on training set:{:.3f}".format(model.evaluate(X_train,y_train)[1]))
print('\033[93m'"Accuracy on Validation set:{:.3f}".format(model.evaluate(X_val,y_val)[1]))
print('\033[92m'"Accuracy on test set:{:.3f}".format(model.evaluate(X_test,y_test)[1]))

Train on 63882 samples, validate on 7985 samples
Epoch 1/2
63882/63882 [==============================] - 8s 128us/step - loss: 0.2330 - accuracy: 0.9359 - val_loss: 0.1844 - val_accurac
y: 0.9390
Epoch 2/2
63882/63882 [==============================] - 8s 125us/step - loss: 0.1819 - accuracy: 0.9393 - val_loss: 0.1826 - val_accurac
y: 0.9388
63882/63882 [==============================] - 4s 66us/step
Accuracy on training set:0.939
7985/7985 [==============================] - 1s 64us/step
Accuracy on Validation set:0.939
7986/7986 [==============================] - 1s 63us/step
Accuracy on test set:0.937
```

Accuracy of predicted values

OBJECTIVE

Reserve Bank of India defines NPA as any advance or loan that is overdue for more than 90 days. “An asset becomes non-performing when it ceases to generate income for the bank,” said RBI in a circular form 2007. To be more attuned to international practises, RBI implemented the 90 days overdue norm for identifying NPAs has been made applicable from the year ended March 31, 2004.

In fiscal year 2019, the value of non-performing assets of private banks across India amounted to over 1.8 trillion Indian rupees. This was much lesser in fiscal year 2017, amounting to about 9.3 billion rupees. Non-performing assets have posed a big problem for banks in India and experts point that this crisis had been long in the making. Since more banks are facing a problem of risky or non-performing assets, the profitability and solvency of banks has gone down.

The problem of NPAs in the Indian banking system is one of the foremost and the most formidable problems that had impact the entire banking system. Higher NPA ratio trembles the confidence of investors, depositors, lenders etc. It also causes poor recycling of funds,which in turn will have deleterious effect on the deployment of credit. The non-recovery of loans effects not only further availability of credit but also financial soundness of the banks.

NPA has the following impact on the bank:

Profitability: NPAs put detrimental impact on the profitability as banks stop to earn income on one hand and attract higher provisioning compared to standard assets on the other hand. On an average, banks are providing around 25% to 30% additional provision on incremental NPAs which has direct bearing on the profitability of the banks.

Asset (Credit) contraction: The increased NPAs put pressure on recycling of funds and reduces the ability of banks for lending more and thus results in lesser interest income. It contracts the money stock which may lead to economic slowdown.

Liability Management: In the light of high NPAs, Banks tend to lower the interest rates on deposits on one hand and likely to levy higher interest rates on advances to sustain NIM. This may become hurdle in smooth financial intermediation process and hampers banks'business as well as economic growth.

Capital Adequacy: As per Basel norms, banks are required to maintain adequate capital on riskweighted assets on an ongoing basis. Every increase in NPA level adds to risk weighted assets which warrant the banks to shore up their capital base further. Capital has a price tagranging from 12% to 18% since it is a scarce resource.
Shareholders' confidence: Normally, shareholders are interested to enhance value of their investments through higher dividends and market capitalization which is possible only when the bank posts significant profits through improved business. The increased NPA level is likely to have adverse impact on the bank business as well as profitability thereby theshareholders do not receive a market return on their capital and sometimes it may erode their value of investments. As per extant guidelines, banks whose Net NPA level is 5% & above are required to take prior permission from RBI to declare dividend and also stipulate cap on dividend payout.

Public confidence: Credibility of banking system is also affected greatly due to higher level NPAs because it shakes the confidence of general public in the soundness of the banking system. The increased NPAs may pose liquidity issues which is likely to lead run on bank by depositors. Thus, the increased incidence of NPAs not only affects the performance of the banks but also affect the economy as a whole. In a nutshell, the high incidence of NPA has cascading impact on all important financial ratios of the banks viz., Net Interest Margin, Return on Assets, Profitability, Dividend Payout, Provision coverage ratio, Credit contraction etc., which may likely to erode the value of all stakeholders including Shareholders, Depositors, Borrowers, Employees and public at large.

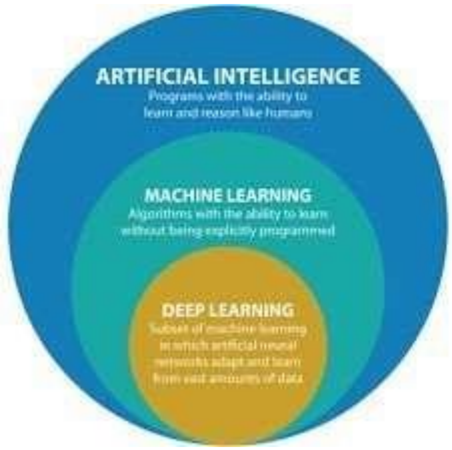
The create a type of deadlock in the banking system in which new customers will not come easily to the bank because of low interest rate provided and old customers will not take the loan as it’s interest rate is high which lead to decrease in the economy as bank plays a major role.

Therefore to prevent the bank from getting bankrupt or increase the profit of the bank we have build a deep learning model which will predict that a customer will repay it’s loan or not.

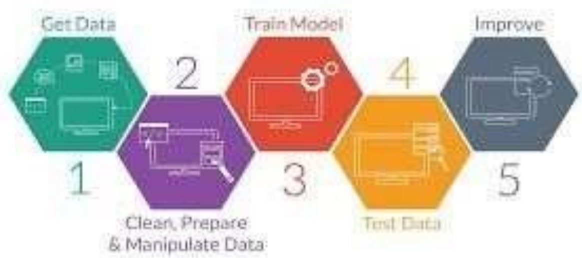
Methodology/Technology or any specific tool to be used

- Deep Learning

Deep learning is an Artificial Intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of Machine Learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network. Deep learning is an AI function that mimics the workings of the human brain in processing data for use in detecting objects, recognizing speech, translating languages, and making decisions. Deep learning AI is able to learn without human supervision, drawing from data that is both unstructured and unlabeled.



In this project we use Deep learning terminologies i.e. Libraries (Keras), Algorithms, Deep Neural Network etc. We take the train dataset data which is used to train our model, when our model is trained then we test our trained model on test dataset to get predictions. After that we check the accuracy of our predicted values, then we decide that is our model is ready and good enough for deployment in real life use. For our Deep Learning model we use activation function Relu.



Hardware Software requirements

1. Anaconda
2. Python 3.7 and its Libraries
   - Tensorflow
   - Sklearn
   - Keras
   - Pandas

- Numpy
- Matplotlib

References

1.https://towardsdatascience.com/common-machine-learning-programming-errors-in-python-
5d76de85e975
2.https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classificationin-python

Contribution

Entire Presentation and collecting data set has been covered by Abhishek.
Technical Portion (Coding) has been covered by Vikrant and Satish.